

# A Different Definition of Success

*an interview with Patrick Conley, Abraxas Software, Inc.,  
by Dave Thompson, Micro Cornucopia*

***If you can't get through the first two years of poverty, don't even think about starting a business. It's called paying your dues; and unless you go through it, you'll never learn to do great things with very little.***

*Another in our "Project Waterfall" series of articles intended to show would-be publishers the perils—and rewards—of software marketing, this interviewee reminds us of the importance of personal motives in judging success.*

**Y**ou have been self-employed for quite a while. How did you wind up starting Abraxas Software?

In our system, the lack of control an individual has over his own life has always frustrated me. I realized at an early age that if I was going to be truly free I had to own my own business. Simple things like what time you get up, how much vacation time you take, how you spend your time, and where you live really are the important things in life. In 1983, during the venture capital rage, I left a lucrative job as a software engineer designing graphics systems and started a company building graphics boards for the PC. Venture capital came very easily and within one month my partner and I had a business, Microfield. Then I learned my first lesson, that venture capitalists were the worst bosses in the world. Taking money from them is like taking money from the devil.

Second, I learned that what Howard Hughes used to say is true: "Partners are dead weight." In a small business you can only have one person in charge. The advantage small businesses have over the big companies is that they react quickly. However, if you have to spend all your time in meetings, you

might as well kiss your business goodbye. When working with VCs, business partners must stick together. But like all strategists, VCs know the best way to control is to divide and conquer.

From 1981 to 1983, I designed graphics controllers at Tektronix, Metheus, and lastly Microfield. That's when I learned my third lesson. At Tektronix, forty of us were developing a system. What a mess: all the people wanted to do was sit in meetings and eat croissants. The project took two years. At Metheus, four people did the same job in half the time. At Microfield, one person did it all in six months. The lesson: One person can do a heck of a lot, something Alexander the Great taught his men. Well, it wasn't long before the VCs wanted me out of Microfield, so they bought me out. With the cash from that venture, I started Abraxas in 1984.

I printed business cards, sold my house (zero gain), got rid of my car payment, and for the next two years I lived on \$500 a month—at Metheus I had been making \$5,000 a month.

The first year and a half I did royalty work, averaging about fifty cents an hour for my time. After that I be-



## A Different Definition...

came a consultant and earned over \$100K a year.

In 1987, I brought out the first product by Abraxas: PCYACC. When I was designing graphics controllers I had been using YACC to build C compilers for Bit Slice engines. In fact, it was at Microfield that I realized the PC market needed a decent YACC for professionals.

If you can't get through the first two years of poverty, don't even think about starting a business. It's called paying your dues, and unless you go through it you'll never learn to do great things with very little.

**Describe Abraxas Software (what it's selling, why, how it is unique, why the products are successful, how they are marketed, how they are supported).**

Right now we sell PCYACC, MACYACC, and PCYACC/2 for OS/2. Later this summer we have some exciting products that will solve portability and maintainability problems. Our products sell because of their value and depth. There are over ten years of work in what we call PCYACC, and at \$395 it's an incredible value for someone serious about language design.

We publicize our products via press releases and advertising. We support our users in terms of helping them solve problems with the program; we don't teach people how to use PCYACC. In fact, the only reason we charge \$395 is so we don't have to deal with amateurs. At the \$99 price level the customer will drive you crazy. When I was developing commercial products, I once spent an hour with a customer only to find that he had the disk upside down in his drive. Believe me, that can generate strange results if you're debugging floppy disk controller drivers over the telephone.

Now PCYACC—our personal version—at \$139, generates two percent of sales, yet accounts for eighty percent of the support. If I were smart I wouldn't even sell the product, but I don't want to lock out the students and the little guy. In the future, the best innovations will come from the guy in some room somewhere in the middle of nowhere. Don't believe

Microsoft when they say that the days of opportunity are gone for the little guy; this simply isn't true. The truth is that the big guys are terrified of small entrepreneurs. [See "Bill Gates at the Boston Computer Society", *this issue*. -Ed.]

I assume that our customers are smarter than we are; you can't fool them. We give inexpensive or free upgrades; almost every suggestion from a customer gets implemented into our products. One of the things that bothers me about our industry is what I call "pet rock" software: junk software that just sits on people's shelves. A lot of people think they can just spend three months on a program, sell it, and rake in the dough. Well, the joke's on them. Remember, the average software entrepreneur never even pays for his PC, let alone makes any money.

**Where do you see the company going (literally and otherwise) in the next few years?**

Abraxas is committed to developing high quality software development tools for DOS, OS/2, Mac, Sun, AIX, VMS, and NeXT. We want all of our tools to be portable to every platform. In the 90s the most important thing will be portability; no longer will firms invest millions of dollars on systems that can only be marketed for one platform. The dinosaurs who don't see this will simply become extinct.

Sometime in the future, we will move the company to a smaller and more livable town. The politicians are now trying to sell Oregon and make it more like California (in terms of L.A.), so we'll probably be moving to eastern Oregon.

**How well is Abraxas doing financially?**

We are not a software publishing company—we are a software development company. I believe that only publishers can make money on software, and to be a publisher you need to be doing at least five million dollars a year. Otherwise you can't do economy of scale, mass mailings, cheap advertising, and low-cost duplication and printing. We take money in and money goes out, but I

still only pay myself \$500 a month because I know how to live on that much money. About seventy-five percent of our revenue goes towards advertising. A publisher probably spends about fifteen to twenty percent. So, we're staying in business, but nobody is getting rich. If you want to be rich, become a publisher.

At the level we advertise, print, and duplicate we have to pay top dollar. A special mention regarding advertising: The big companies get great breaks that big magazines claim don't exist. I think crossing this advertising cost wall is the biggest obstacle to becoming a big company. However, at this time I want to keep Abraxas small, because small companies are fun and you get to work with the customer. In a big company, you spend all of your time dealing with financial and managerial problems.

**Why do you consider Abraxas successful?**

We're not trying to rip anyone off. We sell quality products that we use ourselves. Every product I'll ever sell will be something that I wished I had myself to solve a particular software engineering problem.

Now, let me clarify something: Success cannot be measured financially. Money is simply the blood of a business. If the patient is low on blood, he dies. Too much money is not good, either. The politicians would take your blood if they could; they *will* take your money. Furthermore, if you're rolling in cash, some company like Microsoft will come and take your business away before you know what's happened. There are probably a hundred times as many companies out there that would rather copy products than innovate and they have full-time people just looking for fat little niches.

Lastly, the purpose of this business is to stay in the game. If you run out of revenue, you're out. Playing the game is fun if you like to travel, meet people, and have full control of your life.

(continued on page 37)



## A Different Definition...

(continued from page 32)

The most important reason I started the business was because I believe that two weeks of vacation a year is ridiculous. Since my first year in business, I have taken at least two to three months a year off for vacation. I take very inexpensive trips, but I go all over the world. I try to go mountain climbing every weekend, I ride my bicycle to work, and I do martial arts in the evening. Another thing I like most about my business is buying new toys. I just bought an Apple SE/30. It's a beautiful machine. Soon I'm going to get a NeXT.

One new exciting thing about business is that, as an entrepreneur, you can do something. Give your money away. Yvon Chounaird of Patagonia gives away ten percent of his company profits and the rest goes back into the business. He donates this money to environmental causes. He will not relocate his business to a town that accepts strip malls. If we had more people like this, America would be a place to live instead of a place to work. There's a book called *Growing a Business* that I think everyone should read. It describes many things entrepreneurs can do to make this a better world. Currently, I'm involved in supporting National Public Radio, boycotting EXXON, and saving our environment in general.

### **Why have you been successful while others fail?**

The principle reason, I believe, is this copying tendency we have as Americans. We are rewarded in school for behaving the same way. To be popular, you must dress and act acceptably in school. So when an entrepreneur starts a business, he has a tendency to copy big business. This is a big mistake. You must start lean and mean and never leave that path. I've seen small companies take their first \$100K in venture capital and spend it on furniture so they would feel like executives. Two years later, the same people don't even have a product. If you get caught up in your ego, the fancy car, and all the fluff that goes with being an "entrepreneur," you *will* fail. The first few years you must work long and hard for no return. All things in life are this way. As in mountain

climbing and martial arts, the paths are long but the rewards are great.

The second reason others fail is lack of patience and commitment. There is no quick way to get rich. If that's what you want, become an investment banker and steal legally from the public. You must be committed, burn your bridges, and quit your comfortable job. Make your business the most important thing, after the people in your life. Many people are afraid to do things alone so they bring a friend. We usually spend less time picking a partner than picking a lover; but remember, partners must spend more time together. When the partnership goes on the rocks, the business fails and most people are afraid to go it alone—entrepreneurs must learn to be self sufficient. Misery loves company and most people do not like to suffer alone.

Also, don't ever announce vaporware and don't ship a product with bugs. It only takes one mistake, one message that you are a flaky company, and that's it. No ifs, buts, or anything. It's the old story, "If you don't have time to do it right the first time, where are you going to get time the second time?" I see this over and over. If you ever make this mistake, go back to square one and rename your company because the public will never forgive you.

Lastly, pet rock software: I have seen so much stupid software that it sickens me. The public is not stupid. If you can't spend a long time and develop an excellent product, then don't even bother. If you develop a high-quality, useful piece of software that solves real problems, really works, and does not create more problems, then you cannot possibly fail.

### **You use remote programmers. Why? Tell us about them.**

When I started my business I knew I did not want to build a home for orphan programmers. I have known so many "programmers" who spend their day reading the UnixNet, or playing computer games—I just did not want to be a baby-sitter. Remote programmers, by their very nature, must be disciplined or they simply will not eat. I prefer to hire first-

generation immigrants. They are the hardest workers. They understand that, in terms of starting their own business, America is without limits and they want to take advantage of this.

Everyone who works with me is being trained to have his or her own business. It really bothers me that second-generation people don't want to do what it takes. This will be the competitive downfall of our country. Since I'm on this subject, I want to mention that most of PCYACC sales come from northern Europe. Here we have one of the most technical products available for the PC, and only disciplined North European developers are buying it. This tells me that sometime in the not-too-distant future, Europe is going to blow us away.

One person I work with in New Mexico has a wonderful lifestyle in an enchanting village. He is an incredible producer. When he is not programming for profit, he's teaching his five-year-old son theoretical physics. He has developed an asteroid game that teaches Lorentz transformations, Euclidean geometry, general relativity, and astrophysics—all on a Macintosh. He also has developed technology that allows the Mac to be an inhouse bird sanctuary by playing songs and showing real birds.

### **What sort of advice do you have for other entrepreneurs?**

Work hard. Be original, and ask yourself every day, "How badly do I want this?" Don't abandon the people you love, and, in the end, you're doing it all for only one person—yourself.

Play hard and work hard. Don't quit playing, because if you do you'll lose your sense of value, and then you have nothing. Remember to give away your best ideas. Being selfish with your products and talent will only hurt you in the end.

Every three years take everything you've learned and throw it away. Today, DOS is hot. Eight years ago, it was CP/M. In two years, OS/2 will be hot. Learn the NeXT: Its hardware theme will be the basis of the nineties. Become a registered



## A Different Definition...

developer for all the hardware people like IBM, Apple, HP, Sun, NeXT, and DEC. This way you will be in the same league as the big guys in terms of information and prices on new technology.

### Why do so many software authors struggle or fail?

That's a good question. Most of the people I know are successful or else they get out. My personal experience with our customers is that when they first come out with a great product, it sells well and they pay for their advertising and everything is great. Then, about six months after introduction, they start spending all their time doing support; they never get back to upgrading the product and the revenue decreases. Now they are trapped. They have no time to develop new products. They get squeezed and die. I have seen this happen to many companies in the industry. It is simply bad planning. First of all, most products have a life of only a couple of years. So, it makes sense that you should be developing new products all the time. Also, if your customer base is small, you need to sell upgrades to maintain your cash flow (the blood of a company). You really need at least two people—one to market and one to implement.

Marketing is a full-time job. I spent one year and \$150 thousand dollars of my money marketing PCYACC. I mean, that's all I did: press releases, surveys, talking to customers, solving problems, and more press releases. So you can see what happens when you have one person who tries to do it all. To succeed (stay in business) you must be able to concurrently develop and market. Otherwise, you will fail. This is a hard problem, and I don't have a solution. Abrams solved it because when we introduced PCYACC, we were rolling in cash because of years of consulting jobs. That allowed me to hire great programmers to implement our ideas and I could spend most of my time marketing software.

This reminds me of what Mitch Kapor once said: "All you need to start a software company is a garage, a computer, and five million dollars." That's how he did it.

Now, getting back to being lean and mean. There are ways to market without spending money; you simply must view the world as if you had none. Early on, the solutions will manifest themselves. No one thing makes the human brain operate more effectively than hunger.

Marketing is the creating of a need. Sales is the satisfaction of the need, or delivery. Small software developers usually depend on mail order sales. They pick up the phone and take VISA numbers—that's sales. It's easy, right? However, first you must convince people that they need your product. If the product is new and they've never heard of it before, it costs more, or takes longer, to create that need.

### What would you recommend?

Work hard, be realistic, and live like a pauper. Expect and be prepared for the worst, and pray for the best.

I suspect it's best to publish your own software. To a point, it really depends on your resources and the product target. It would be ridiculous, for instance, to have a small company bring out a better word processor. You're better off licensing it to a big company that needs one for their product mix.

In general, it's better for small companies to go after new niches. The people I know who do the best have their own niche sewed up.

There are three ways to get paid in this business: stock, royalty, and cash. Stock is a joke ninety-nine percent of the time. It is worthless and always will be. Royalty is only good if your product is published by a sincere publisher who will be marketing only your product that season. Cash is always the best in any case. Have you ever noticed that's what lawyers, doctors, and accountants take? Companies only pay royalty for one reason: They can't afford to front the cash. If they really believed in the product, they would never do royalties, because they're profits. If the product does sell well, they'll do everything possible to void the royalty contract and keep the money for themselves. There's no risk for com-

panies taking royalty jobs. The developer has all the risk.

### Why don't they take your advice?

I really believe most people want to be in business for the form and not substance. When people start a business, they're always haggling over business cards, stationery, going public, titles, offices, furniture, travel, expense accounts, and clothes. Notice that I didn't mention product. Most people, I have found, are only interested in the "image" of business. To be successful, you must be product-oriented. Every moment must be spent on product improvement and sales. Every dollar must be spent on increasing sales. Every minute during the day, I ask myself Townsend's question (the guy who rescued AVIS in the sixties), "If you're not doing it for fun or profit, then why the hell are you doing it?"

Now unless folks think incorporating is fun, and I'll tell you it definitely is not profitable in and of itself, then don't worry about it. Simply worry about making your business real. Create a good product.

### If you were starting over, how would you do it differently?

I absolutely would not do royalty work unless the publisher was committed to marketing my software. Also, the publisher would have to guarantee to spend at least three times my guaranteed royalties on advertising.

I guess I had to learn for myself that you must never take anyone's money when you start a business, because if you do, the business is not really yours, even when you do all the work. I learned my lesson quickly, and even got some cash out of the deal. Some people don't learn this lesson until it's too late. Life is full of lessons that we can learn from, or that we can ignore and repeat. If you're really smart, you can listen and watch others and eliminate your own suffering.

Copyright © 1989, Micro Corporation.